



# Génération de Tests pour les Systèmes Temps-Réel à partir de Modèles SysML: Contexte, Motivations et Objectifs

Jean-Marie Gauthier

## ► To cite this version:

Jean-Marie Gauthier. Génération de Tests pour les Systèmes Temps-Réel à partir de Modèles SysML: Contexte, Motivations et Objectifs. CIEL 2013, 2ème Conférence en Ingénierie du Logiciel, Jan 2013, France. hal-00956579

**HAL Id: hal-00956579**

**<https://hal.science/hal-00956579>**

Submitted on 6 Mar 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Génération de Tests pour les Systèmes Temps-Réel à partir de Modèles SysML : Contexte, Motivations et Objectifs

Jean-Marie Gauthier

FEMTO-ST DISC, Besançon, France  
jean-marie.gauthier@femto-st.fr

## Résumé

Cet article présente une approche de génération de tests à partir de modèles SysML pour les systèmes embarqués temps-réel. Ces travaux se basent sur une chaîne outillée existante qui a été proposée dans le cadre du projet de recherche VETESS qui visait la génération de tests pour les systèmes embarqués automobiles. Dans cet article, nous identifions les axes d'amélioration permettant de consolider et de faire évoluer cette approche initiale notamment par la prise en compte des aspects temps-réel. Ces axes d'amélioration, présentés sous la forme de questions de recherche, ont pour objectif de proposer une solution de génération de tests outillée pour les systèmes embarqués temps-réel. Finalement, l'approche proposée est brièvement positionnée au sein de travaux connexes du domaine.

**Mots-clés :** Génération de Tests, Modèle SysML, Systèmes Embarqués Temps-Réel

## Abstract

This paper presents an approach of Model-Based Testing for real-time and embedded systems using SysML. This work is based on an existing approach that has been proposed during the VETESS project. This approach aims to generate tests for embedded systems. In this paper, we identify areas of improvement, which permit to evolve the initial approach by taking into account the real-time aspects. These areas, which are presented in the form of research questions, aim to propose an automated Model-Based-Testing toolchain for real-time and embedded systems. Finally, the proposed approach is briefly included in the related works.

**Keywords :** Tests Generation, SysML Modeling, Real-Time and Embedded System

## 1 Introduction

La validation des systèmes embarqués temps-réel est une activité complexe car ces systèmes sont souvent composés d'une partie matérielle et d'une partie logicielle possédant des contraintes temporelles. Cette hétérogénéité rend difficile la validation de ce type de système au regard de leurs exigences fonctionnelles et non-fonctionnelles. Pour répondre à ce défi, nous proposons de mettre en place un processus de génération de tests pour les systèmes embarqués temps-réel sur la base des résultats obtenus dans le cadre du projet VETESS<sup>1</sup> (Vérification de systèmes embarqués VEhicules par génération automatique de TESTs à partir des Spécifications). Ce processus outillé, basé sur l'Ingénierie Dirigée par les Modèles, permet de vérifier et de simuler des modèles de systèmes dès la première étape de la conception. C'est dans ce contexte que

---

1. <http://lifc.univ-fcomte.fr/vetess/>

s'inscrivent nos travaux réalisés dans le cadre du projet SyVAD <sup>2</sup> (Modélisation SysML pour la Validation d'Applications Distribuées). Cet article présente, sur la base des travaux antérieurs de l'équipe, les objectifs de nos travaux et leurs perspectives.

Cet article se décompose en quatre parties. La Section 2 introduit l'approche existante, issue du projet VETESS, de génération de tests pour les systèmes temps-réel. La Section 3 met en exergue les contributions et les extensions que nous souhaitons apporter à cette approche. Des travaux connexes sont présentés dans la Section 4. Finalement, la Section 5 conclut et présente les perspectives dans le contexte du projet SyVAD.

## 2 Synthèse des travaux antérieurs

Les travaux issus du projet VETESS ont abouti au développement d'une chaîne outillée permettant la génération de tests à partir de modèle UML/SysML pour les systèmes embarqués temps-réels [11]. Cette chaîne outillée, illustrée par la Figure 1, permet la génération de tests abstraits et leur concrétisation afin de les exécuter sur un banc de test réel ou sur un simulateur.

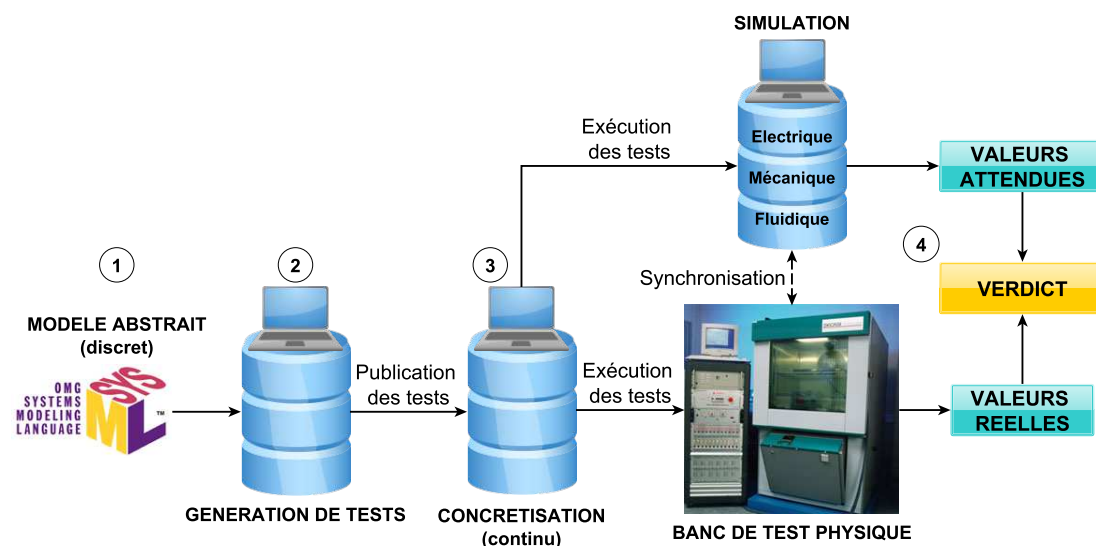


FIGURE 1 – Chaîne outillée existante

La première étape de cette approche (①) consiste à modéliser le système sous test en utilisant un sous-ensemble du langage SysML baptisé SysML4MBT [4]. Ce sous-ensemble contient le diagramme de définition de bloc pour représenter la vue statique du système (blocs, associations, compositions, énumérations, attributs de bloc, opérations, signal et port de flux), le diagramme interne de bloc pour représenter les interactions entre les blocs et le diagramme d'états-transitions pour représenter l'aspect dynamique du système. De plus, le diagramme d'exigence permet d'explicitier les exigences du système afin d'assurer une traçabilité bidirectionnelle entre les tests générés et les exigences initiales.

Finalement, ce modèle est rendu formel par l'ajout de contraintes exprimées avec un sous-ensemble du langage OCL baptisé OCL4MBT [4], permettant ainsi d'automatiser la production des cas de test à partir de tels modèles. Le sous-ensemble SysML4MBT permet une modélisation

2. <http://syvad.univ-fcomte.fr/>

qui s'appuie sur une sémantique opérationnelle événementielle du système sous test. Cette modélisation permet alors de discrétiser le système et son environnement, c'est-à-dire, que le modèle spécifie la situation du système à des moments précis (à la réception d'un stimulus) sans étudier les variations entre deux états stables.

Durant la seconde étape (②), l'outil de génération de tests *Test Designer<sup>TM</sup>* [3] permet de générer des tests abstraits en appliquant un critère de couverture de type structurel (couverture d'états, transitions, etc). Ces tests sont dits abstraits car ils sont exprimés avec les éléments du modèle *SysML4MBT*.

La troisième étape (③), appelée concrétisation, consiste à donner des valeurs concrètes aux éléments abstraits des cas de tests abstraits générés [12] (données et opérations). Durant cette étape, il est nécessaire de préciser le temps qui s'écoule entre deux stimuli afin d'obtenir une séquence de tests exécutable sur le système réel. L'aspect continu du système est donc réintroduit au niveau de cette étape.

Finalement (④), les séquences de tests sont exécutées sur le système réel et simulées dans un environnement *Matlab* [13]. Les valeurs issues de la simulation sont alors comparées avec les valeurs réelles afin de donner un verdict aux cas de test.

Cette approche outillée a montré sa pertinence pour générer automatiquement un nombre maîtrisable de cas de test tout en assurant une couverture optimale du modèle. De plus, celle-ci offre la capacité de régénérer et de ré-exécuter automatiquement les cas de test pour toute modification du modèle. Les expérimentations ont montré que le temps de génération des cas de test est faible en comparaison du temps passé à écrire le modèle. En effet, la complexité des modèles étant raisonnable, les outils de génération de tests sont maintenant relativement matures pour être efficace en terme de temps de génération et de taux de couverture du modèle. Les retours d'expérience obtenus lors de ce projet ont ainsi été jugés très satisfaisant par les ingénieurs de PSA qui ont validé ces résultats dans le cadre du projet VETESS [2].

Toutefois, la mise en œuvre de cette approche a montré certaines faiblesses et les études de cas réalisées ont permis d'identifier plusieurs points d'amélioration :

1. L'ingénieur validation est obligé de spécifier à la fois le modèle *SysML4MBT* et le modèle de simulation *Matlab*. De plus, la validation fonctionnelle du modèle *SysML4MBT* n'est effective qu'au moment de l'exécution des tests sur le modèle de simulation. Cette validation tardive du modèle *SysML4MBT* retarde la mise au point du modèle et le cycle de stabilisation du modèle peut être coûteux en temps et en effort.
2. Le modèle *SysML4MBT* s'appuie sur une discrétisation du système réel. Ainsi les aspects continus du système sous test sont pris en compte durant la phase de concrétisation des tests. La couche d'adaptation doit être corrigée à chaque modification du modèle ce qui engendre une maintenance et une stabilisation du code très coûteuse qui nuit au caractère automatique de la solution proposée et limite la réutilisation d'un projet à un autre.

Cette section a présenté une synthèse des travaux antérieurs dans le contexte de la génération de tests basée sur les modèles pour les systèmes embarqués temps-réels. Ces travaux mettent en lumière deux pistes principales d'améliorations qui motivent directement l'axe de nos recherches de doctorat : capturer les aspects temps-réel au sein du modèle *SysML4MBT* et les propager de façon automatisée dans le reste de la chaîne. Ces pistes doivent permettre de :

- faire du modèle *SysML4MBT* l'unique référentiel fonctionnel,
- réduire les efforts et accélérer la phase de validation fonctionnelle,
- et d'étendre les algorithmes de calcul de tests avec la dimension temps-réel.

### 3 Approche proposée

Sur la base des conclusions du projet VETESS introduit en section précédente, nous présentons les directions de recherche qui consistent à prendre en compte l’aspect temps-réel du système sous test dès la phase de modélisation. Cet objectif pose alors les problématiques de recherche suivantes :

1. Comment valider le modèle **SysML4MBT** en amont de la phase d’exécution des tests sur le modèle de simulation ?
  - Éviter l’écriture d’un modèle **SysML4MBT** et d’un modèle de simulation.
  - Valider fonctionnellement le modèle en amont de la génération de tests.
2. Dans quelles mesures peut-on prendre en compte le caractère temps-réel du système dès la phase de modélisation afin de l’exploiter tout au long du processus de génération automatique de tests ?
  - Renforcer la pertinence des cas de tests générés pour les aspects temps-réel.
  - Améliorer et stabiliser l’automatisation du processus de concrétisation.

Pour répondre à la première problématique, l’utilisation des techniques issues de l’Ingénierie Dirigée par les Modèles permettrait d’effectuer une transformation automatique depuis le modèle **SysML4MBT** pour produire le modèle de simulation. Pour mettre en place cette transformation, il est nécessaire de prendre en compte les équations du système sous test dès la phase de modélisation, ce que **SysML** offre à travers le diagramme paramétrique. Une extension du langage **SysML4MBT** est donc envisagée pour prendre en charge ce diagramme. De plus, une telle transformation permettrait de valider le modèle **SysML4MBT** plus tôt dans la chaîne outillée et de garantir la cohérence fonctionnelle des tests avant leur exécution sur le système réel.

Afin de prendre en compte l’aspect temps-réel du système sous test, l’utilisation du profil **MARTE** [15] de l’OMG semble être une piste intéressante pour compléter le modèle **SysML4MBT**. En effet, le profil **MARTE** permet l’introduction des aspects temps-réel au sein d’une modélisation **SysML**. En donnant une sémantique interprétative non ambiguë au profil **MARTE**, nous serions alors en mesure de spécifier l’intervalle de temps qui s’écoule entre deux stimuli permettant de minimiser l’effort de concrétisation. En outre, il serait également possible de proposer de nouveaux critères de couvertures basés sur les aspects temporels pour générer les cas de test.

Finalement, afin d’améliorer l’automatisation du processus de concrétisation, nous envisageons d’utiliser le profil **UTP** [16]. Ce profil permet d’exprimer les artefacts du processus de génération de tests de manière standardisée. Ainsi, en utilisant les techniques de transformation de modèles, nous pourrions automatiser la publication des tests et leur concrétisation en utilisant **UTP** comme un langage pivot.

La prise en compte des différentes pistes d’amélioration citées précédemment, nous amène à revisiter le processus et la chaîne outillée proposée initialement par le projet VETESS. Cette nouvelle approche pour les systèmes embarqués temps-réel est illustrée par la Figure 2. La section suivante présente des travaux connexes sur lesquels nous allons nous appuyer pour faire émerger cette solution.

### 4 Travaux Connexes

Outre les travaux antérieurs cités au début de cet article, nous avons identifié des résultats dans les différents domaines dont nous pouvons nous inspirer pour nos futurs travaux. Les travaux de RTaW [17] proposent une transformation de modèle **SysML** vers le langage de simulation **VHDL-AMS** [6]. Cette transformation prend en compte le diagramme de définition de bloc, le diagramme interne de bloc et le diagramme paramétrique pour générer du code **VHDL-AMS**.



## Remerciements

Je souhaite remercier Fabrice Bouquet, Ahmed Hammad et Fabien Peureux pour les relectures et les conseils apportés lors de la rédaction de cet article. Ce travail est supporté par le Conseil Régional de Franche-Comté avec le projet SyVAD (<http://syvad.univ-fcomte.fr/syvad/>).

## Références

- [1] Acceleo Documentation. <http://www.eclipse.org/acceleo/>, last viewed december 2012.
- [2] F. Ambert, F. Bouquet, J. Lasalle, B. Legeard, and F. Peureux. Applying an MBT Toolchain to Automotive Embedded Systems : Case Study Reports. In *VALID'12, 4-th Int. Conf. on Advances in System Testing and Validation Lifecycle*, pages 139–144, Lisbon, Portugal, November 2012.
- [3] F. Bouquet, C. Grandpierre, B. Legeard, and F. Peureux. A test generation solution to automate software testing. *Proceedings of the 3-rd Int. Workshop on Automation of Software Test (AST'08)*, pages 45–48, May 2008.
- [4] F. Bouquet, C. Grandpierre, B. Legeard, F. Peureux, N. Vacelet, and M. Utting. A subset of precise UML for model-based testing. In *Proceedings of the 3-rd Int. Workshop on Advances in Model Based Testing (A-MOST'07)*, pages 95–104, London, UK, July 2007.
- [5] J. Bézivin, G. Dupé, F. Jouault, G. Pitette, and J. Eddine Rougui. First experiments with the ATL model transformation language : Transforming XSLT into XQuery. In *2-nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture*, 2003.
- [6] E. Christen and K. Bakalar. VHDL-AMS-a hardware description language for analog and mixed-signal applications. *IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing*, 46(10) :1263–1272, October 1999.
- [7] J-M. Gauthier, F. Bouquet, A. Hammad, and F. Peureux. Transformation of SysML structure diagrams to VHDL-AMS. In *IEEE Workshop on design, control and software implementation for distributed MEMS (dMEMS'12)*, Besançon, France, 2012. IEEE CPS.
- [8] J-M. Gauthier, F. Bouquet, F. Peureux, and A. Hammad. Verification and Validation of Meta-Model Based Transformation from SysML to VHDL-AMS. In *Proceedings of the 1-st Int. Conf. on Model-Driven Engineering and Software Development (MODELSWARD'13)*, pages 123–128, Barcelona, Spain, February 2013.
- [9] M. Iqbal, S. Ali, T. Yue, and L. Briand. Experiences of Applying UML/MARTE on Three Industrial Projects. In *Model Driven Engineering Languages and Systems*, volume 7590 of *Lecture Notes in Computer Science*, pages 642–658. Springer Berlin Heidelberg, 2012.
- [10] B. Pérez Lamancha, P. Reales Mateo, I. Rodríguez de Guzmán, M. Polo Usaola, and M. Piattini Velthius. Automated model-based testing using the UML testing profile and QVT. In *Proceedings of the 6-th International Workshop on Model-Driven Engineering, Verification and Validation (MoDeV'09)*, pages 1–6, Denver, CO, USA, October 2009.
- [11] J. Lasalle, F. Peureux, and F. Fondement. Development of an automated MBT toolchain from UML/SysML models. *Innovations in Systems and Software Engineering*, 7 :247–256, 2011.
- [12] J. Lasalle, F. Peureux, and J. Guillet. Automatic test concretization to supply end-to-end MBT for automotive mechatronic systems. In *Proceedings of the 1-st International Workshop on End-to-End Test Script Engineering, ETSE '11*, page 16–23, Toronto, Canada, 2011.
- [13] Mathworks, 2012. <http://www.mathworks.fr>, last viewed december 2012.
- [14] Mohamed Mussa and Ferhat Khendek. Towards a model based approach for integration testing. In *Proceedings of the 15-th International Conference on System Design Languages (SDL'11)*, pages 106–121, Toulouse, France, July 2011. Springer-Verlag.
- [15] OMG. MARTE. Technical report, ptc/11-06-02. Object Management Group, 2011.
- [16] OMG. UML Testing Profile. Technical report, ptc/12-04-01. Object Management Group, 2012.
- [17] Real-Time At Work. <http://www.realtimeatwork.com>, last viewed december 2012.